



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/772,626	02/05/2004	Michael Kovacs	BEAS-01302US1	5661
23910	7590	02/08/2008	EXAMINER	
FLIESLER MEYER LLP			VU, TUAN A	
650 CALIFORNIA STREET			ART UNIT	PAPER NUMBER
14TH FLOOR			2193	
SAN FRANCISCO, CA 94108				
MAIL DATE		DELIVERY MODE		
02/08/2008		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

MK

Office Action Summary	Application No.	Applicant(s)
	10/772,626	KOVACS ET AL.
	Examiner	Art Unit
	Tuan A. Vu	2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 19 November 2007.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1,2,4-20 and 22-27 is/are pending in the application.
 - 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1,2,4-20 and 22-27 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/ are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date: _____
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)	5) <input type="checkbox"/> Notice of Informal Patent Application
Paper No(s)/Mail Date: _____	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 11/19/07.

As indicated in Applicant's response, claims 1-2, 4-15, 18, 22, 25, 27 have been amended. Claims 1-2, 4-20, 22-27 are pending in the office action.

Claim Objections

2. Claims 1, 2, 4, 5, 8-9 are objected to because of the following informalities: the extensive use of 'capable of' (e.g. capable of *generating, creating, comparing, updating, repairing* etc.) in these claims to describe a functional feature of the invention appears to be weak choice of language. By 'capable of' one would understand that there is potentiality for some function to be performed; and mere reciting of a unrealized capacity - a potential - cannot be viewed as the true extent of what the invention really amounts to or asserts itself in terms of carrying out its functionality. Suggested forms of correction to 'capable of' can be, for example, 'validator to validate' or 'validator operable to validate'. Appropriate correction is required, for fear that a USC § 112, 2nd paragraph type of non-compliancy would have to be looked into.

3. Claims 1 and 8 recites 'automatically repair' and "automatically repairing" a 'defective descriptor'. In light of the lexicographic connotation imparted to any automated mechanism for repair, the above use of language has to entail a programmatic function by which an object is being repaired from a broken state to a repaired state (non defective state). The term 'repair' is not redefined anywhere in the Disclosure to mean otherwise, and absent any specific teaching as to how repairing is done **automatically**, the above phrase is not a correct syntactic construct deemed acceptable by commonly accepted meaning or knowledge. Updating a incorrect piece of information within a tree node with proper or newer information is not same as automatically

repairing a broken (defective) node on a tree. Correction is required for re-adjusting this ‘automatically repairing’ phraseology and in the mean time, the referred to ‘automatic repair’ connotation will be treated as updating in **response to input** provided by some GUI indication.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 1-2, 4-14 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a “useful, concrete, and tangible result” be accomplished. An “abstract idea” when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the “useful arts” when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a “useful, concrete and tangible result”.

The current focus of the Patent Office in regard to statutory inventions under 35 U.S.C. § 101 for method claims and claims that recite a judicial exception (software) is that the claimed invention recite a practical application. Practical application can be provided by a physical transformation or a useful, concrete and tangible result. The following link on the World Wide Web is the United States Patent And Trademark Office (USPTO) reference in terms of guidelines on a proper analysis on 35 U.S.C. §101 rejection.

<http://www.uspto.gov/web/offices/pac/dapp/opla/preognnotice/guidelines101_20051026.pdf>

Specifically, claim 1 recites a computer-based system comprising a parser, a generator, a GUI for deploying an application. There is not sufficient teaching for this claimed ‘system’ to convey existence of hardware support **included** therein to effectuate the functionality of the

parser or the *generator*, all of which being described as mere (computer-based) software entities included in the tool of Figure 2 of the Disclosure. Moreover, there is no sturdy evidence that said software entities actually do externalize any such functionality. That is, in light of a **potent** functionality termed as ‘capable of’ (of *generating, creating, validating, repairing, deploying*), it is even harder to construe that the claim as a whole would be able to realize data transformation leading to any sort of result. Lacking hardware support in order to carry out the functionality as mentioned above, the above software listing amounts to mere ‘Functional Descriptive Material’ which does not really belong to any of the 4 statutory categories (composition of matter, machine/apparatus, process method, article of manufacture), material which further amounts to an abstract idea or a potentiality not realizable via actual real-world data transformation with machine support thereby yielding a concrete, tangible, and useful result. The claim is rejected as non-statutory for not fulfilling a Practical Application result as set forth in the above Guidelines *pdf* file (see ‘Functional Descriptive Material’, section: Annex IV(a), pg. 53-54).

Claims 2, 4-7 are also rejected for not remedying to the hardware support deficiency of the base claims.

Claim 8 recites a computer based system comprising a parser, a generator and a builder, all which software based functional entities described as merely ‘capable of’; and for just reciting Functional Descriptive Material (i.e. without hardware support or tangible embodiment), in the absence of actions being **actually** taken, the claim is likewise rejected as non-statutory as set forth for claim 1. Claims 9-14 are also rejected for not providing hardware support to embody the software entities of the base claims.

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claim 2 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 2 is rejected as being incomplete for omitting essential structural cooperative relationships of elements, such omission amounting to a gap between the necessary structural connections. See MPEP § 2172.01. The omitted structural cooperative relationships are: what GUI utility is being subject to navigation by the system in order for a source of a error to be reached.

The claim recites ‘system can navigate the GUI to the source of the error’, hence does not provide clear structural interrelationship between GUI elements recited (pane, hierarchy, message, toolbar, source of error) as to the act of navigating by the ‘system’ based on user selection of error messages. The ‘navigate’ scenario, in light of the Specifications amounts to highlighting a node (Specs, pg. 6, para 0021) by the method; that is, based on identification stored in the error message when such message is selected, the validator can visually present to the user the field in the so-identified node within the pane. The GUI is an encompassing software environment in which both the pane, and the hierarchy of nodes are presented along with the rest of the browser layout. One of ordinary skill would not see how the GUI can be subject to navigation by any utility based on the highlighting as described above. For one of ordinary skill in the art recognizing the connotation of ‘navigate’, the claim is not supported by a clear structural relationship between hierarchy of node with respect to a broader GUI environment, and when the claim recites ‘navigate the GUI to the source’, it is hard to construe

how the GUI can be navigated, absent a navigation utility anywhere in the Disclosure to enable such navigation to take place. In the extent that the hierarchy is traversed in order for a node to be highlighted, there is clear teaching that not the GUI (emphasis added) but only a mere reading of an identifier (using validator 302) inside a selected message (Specs: para 0020, pg. 6), the remote concept of navigation taken out of context. The ‘navigate the GUI’ is not given weight and for its merits, will be treated as mere **reading** of message content to identify a source error.

8. Claims 12, 18 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention; that is, these claims recite the limitation "modules are not deleted from the first representation". There is insufficient antecedent basis for this limitation in the claim, and this will be treated as mere information.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 1, 4-20, 22-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over WebLogic Server 6.1: *Developing Weblogic Server J2EE Applications*, 10/22/2001, pp. 1-20 <<http://web.archive.org/web/20011022014739/edocs.bea.com/wls/docs61/programming/environment.html>> (hereinafter WLS_6.1).

As per claim 1, WLS_6.1 discloses a system for automatically maintaining at least one deployment descriptor, comprising:

a parser capable of generating a representation (e.g. *Console* – pg. 9 bottom → top pg. 10; *navigate tree* - pg. 14-16 – Note: console including tree enabling editing of a Descriptor reads on generating representation thereof on console) of the at least one deployment descriptor; a generator capable of creating the at least one deployment descriptor (e.g. XML editor, pg. 9-10; pg. 14-16);

a validator capable of validating the at least one deployment descriptor (e.g. *click Validate* – pg. 16, item 10, top; *click Validate* – pg. 17 item 10);
a graphical user interface (GUI) capable of invoking the parser, wherein the GUI include a user-selectable resource hierarchy, settings pane, message area and toolbar (e.g. step 2-6: *browser, drop-down menu, navigate, click, edit text, form, pane* – pg. 15 – Note: browser with pane including tree nodes for users to navigate and click **reads on** selectable hierarchy generated from the invoked parser);

and wherein the system is capable of automatically deploying an application associated with the at least one deployment descriptor (sec *Web Applications* – pg. 16; sec *Enterprise Applications* – pg. 19).

WLS_6.1 does not explicitly disclose wherein the system is capable of automatically repairing a first deployment descriptor of the at least one deployment descriptor if the first deployment descriptor is defective (Note: repairing is treated as updating – see Claim Objections). WLS_6.1 teaches editor using a form of validator to delete some tree entries in order to persist a confirmed or accepted descriptor into a target application (see substep items 9-11, pg. 17; substep items 8-11, pg. 18-19). In view of the role played by a validator to detect a syntactic or rule-infringing error by the construct of the Descriptor, it would have been obvious

for one of skill in the art at the time the invention was made to implement the WLS_6.1 validator in such a way to apply the rule checking as set forth above in order to automatically correct any error or discrepancy (e.g. non-compliant Descriptor being readjusted or updated) by a deployment Descriptor before persisting its corrected version thereof into the Application for deployment, and this is consistent with WLS_6.1 (see pg. 5, substep 7; *Java Compiler, Development Weblogic Server* -pg. 10) utilization of Deployment Descriptor Console to support deploying a enterprise/Web application.

As per claim 4, WLS_6.1 discloses wherein the generator is capable of producing the at least one deployment descriptor from at least one source code file (e.g. XML editor, schema - pg. 14, middle; *web.xml weblogic.xml files* - substep 4, pg. 2).

As per claim 5, WLS_6.1 discloses builder component capable of automatically updating the at least one deployment descriptor to reflect one or more changes (e.g. Using the Administration Console ... Editor ... click the Persist button - pg. 14) in at least one source code file.

As per claim 6, WLS_6.1 discloses wherein the representation can include information pertaining to at least one of: a Java.TM. archive (JAR), a Web Archive (WAR), an Enterprise Archive (EAR), and a Java.TM. Connector Architecture Component (RAR) – see pg. 2, step 4 to step 7; steps 5-7, pg. 5; *Resource Adapter .rar* - pg. 6, pg. 8).

As per claim 7, WLS_6.1 discloses wherein the at least one deployment descriptor can be expressed as an Extensible Markup Language document (refer to claim 4- see *ejb-jar.xml* - pg. 14-15).

As per claim 8, WLS_6.1 discloses a computer-based system for automatically maintaining at least one deployment descriptor, comprising:

a parser capable of generating a first representation of the at least one deployment descriptor; a generator capable of creating a second representation of at least one deployment descriptor based on one or more source files (see substeps 2-5 - pg. 14-15 – Note: XML-parsing with GUI-based tree creation of nodes with each representing a descriptor reads on first and second representation);

all of which having been addressed in claim 1.

WLS_6.1 does not make it explicit that the system is capable of automatically repairing a first deployment descriptor of the at least one deployment descriptor if the first deployment descriptor is defective; but this repairing limitation has been addressed in claim 1.

WLS_6.1 does not explicitly disclose a builder capable of comparing the first representation with the second representation; updating the first representation to create an updated first representation based on the second representation if at least one source file of the first representation is modified, and generating new deployment descriptors from the updated first representation.

However, WLS_6.1 mention about use of a console to create descriptor for deploying EJB or packages to be compiled, which compilation entails version compatibility for the Jar files being deployed using the Console (see pg. 12, middle) as well as verifying version compatibility for non-BEA developers otherwise (see *runtime version* - Web Browser; *software compatibility* - Third Party Software, pg. 11). Since the Jar files encompass retrieval of persisted information for creating deployment descriptor, the necessity as to check on up-to-date version information

for those source files is strongly suggested from the above builder environment, e.g. comparing their respective version-controlled identification. Based on well-known concept that developing code based on files fetched for enlistment in a builder (as purported by WLS_6.1) would require that compatibility of files or properties related to deployment are met such as mentioned above, it would have been obvious for one of ordinary skill in the art to implement WLS_6.1 builder with capability to make use of XML Jar files and ensuring that the derived descriptor are also compatible with the deployment of the J2EE Java applications as contemplated above. One would be motivated to implement an update capability in WLS_6.1 builder using its editor (see Editing Deployment Descriptors – pg. 13-15) such that modified or outdated (with respect to version compatibility as taught above) Gui representation of such derived descriptor would be detected via comparison among their respective versioned identity, wherein the updating step to correct any such out-of-date Descriptor (i.e. update first representation of descriptor based on Jar file being modified, then generating new update descriptors therefor) via the Console as set forth in claim 1, using the repairing capability to readjust such a detected error for the reasons as set forth therein.

As per claim 9, refer to claim 4 for descriptor generated from one source code file.

As per claims 10-11, refer to claims 6-7, for respective rejection.

As per claims 12-13, WLS_6.1 discloses wherein module (i.e. information) is not deleted from the first representation, and information in the second representation that is not in the first representation is added to the first representation (substep 7, pg. 15; *add elements* - substep 5, pg. 16 – Note: adding one element from a second representation to a first representation reads on the latter representation not deleted).

As per claim 14, WLS_6.1 discloses wherein a user can modify information in the second representation (e.g. substep 7, pg. 15; substep 8, pg. 17, pg. 18; substep 5, pg. 19).

As per claim 15, WLS_6.1 discloses a method for updating at least one deployment descriptor, comprising:

creating a first representation of the at least one deployment descriptor;
creating a second representation of a second at least one deployment descriptor based on one or more source files;
all of which being addressed in claim 8.

But WLS_6.1 does not explicitly disclose comparing the first representation with the second representation; and updating the first representation to create an updated first representation based on the second representation if at least one source file of the first representation is modified, and generating new deployment descriptors from the updated first representation. However, the above has been addressed as obvious in light of the rationale as set forth in claim 8.

As per claims 16-17, refer to claims 6-7, respectively.

As per claims 18-19, refer to claims 12-13.

As per claim 20, refer to claim 14.

As per claim 22, WLS_6.1 disclose machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

create a first representation of the at least one deployment descriptor; create a second representation of a second at least one deployment descriptor based on one or more source files; compare the first representation with the second representation; update the first representation to

create a first representation of the at least one deployment descriptor; create a second representation of a second at least one deployment descriptor based on one or more source files; compare the first representation with the second representation; update the first representation to create an updated first representation based on the second representation if at least one source file of the first representation is modified, and generating new deployment descriptors from the updated first representation.

all of which having been addressed in claim 15, incorporating the obvious rationale as set forth therein.

As per claims 23-24, refer to claims 6-7, respectively.

As per claims 25-26, refer to claims 12-13.

As per claims 27-28, refer to claims 20-21.

11. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over WebLogic Server 6.1: *Developing Weblogic Server J2EE Applications*, 10/22/2001, further in view of Reddy et al, USPN: 5,845,120 (hereinafter Reddy)

As per claim 2, the limitation as to generating a GUI-based type of error (by the validator) when it encounters a syntactic or semantic fault in the at least one deployment descriptor, although not explicitly taught by WLS_6.1, has been rendered obvious in light of the rationale in claim 1.

Nor does WLS_6.1 explicitly disclose that the displayed GUI error message is selectable error message such that in response to user's selection of said error message, navigating (by the system) the GUI to the source of the error corresponding to the selectable error message. Error message with underlying information which can be displayed to user upon user's request therefor

has been taught in error validating analogous arts at the time the invention was made. Reddy disclose automated information display in conjunction with parsing a markup language represented by tree of source code constructs (e.g. Fig. 1-2) within a compilation process, enabling user to select specific parts of an error message associated with part of the language syntax being parsed, thereby obtaining the associated information pertinent to said language construct syntax violation (see Summary, col. 2 lines 45-61). Based on the rationale from above by which error message is displayed, the knowledge as to the source of the error would have implied need for the user to be learned about the information underlying the error, i.e. the cause of the syntax violation as set forth in WLS_6.1. Thus, it would have been obvious for one skill in the art at the time the invention was made to implement the error message in WLS_6.1 validating process to include the help information as by Reddy, because when the user can specify which construct has been highlighted as an error, the textual and helpful information as to the real cause/source of the error being provided by the validating system itself, as explained in Reddy's approach, would help provide the user with specific understanding as to the exact nature of syntax violation pertinent to that specific and highlighted portion of the markup construct.

Response to Arguments

12. Applicant's arguments filed 11/19/07 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

Applicants have submitted claim 2(Appl. Rmrks pg. 10, top) includes a selectable error message; but this is a new limitation addressable with necessitated new grounds of rejection. Hence the argument is moot.

The alleged statement that Weblogic 6.1 does not render obvious claim 5, dependent from claim 1 (Appl. Rmrks pg. 10, middle) is deemed a mere assertion against a prima facie § 103 rationale, against which Applicants fail to explain how the mechanics of such rationale would be part by part improper.

The argument in regard to the limitations of claim 8, 15, 22 (Appl. Rmrks pg. 10, bottom) also fall under the ambit of new grounds of rejection being necessitated by the amendments.

The rest of the claims (Appl. Rmrks pg. 11) will fall under the rejection set forth in the base claims; and since no argument is deemed commensurate with the previously submitted claim set, the current grounds of rejection as per this Amendment will stand.

Conclusion

13. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Tuan A Vu
Patent Examiner,
Art Unit 2193
February 06, 2008